

How to Perform Linear Hypothesis Testing in R Using `linearHypothesis()`

Authored by
stats writer

November 22, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Perform Linear Hypothesis Testing in R Using linearHypothesis()*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=99648>

The **linearHypothesis()** function is a cornerstone tool in modern applied statistics using the R environment. It allows researchers and analysts to conduct rigorous linear hypothesis tests on fitted statistical models, particularly those derived from linear models or generalized linear models. This powerful function assesses whether a specific linear combination of regression coefficients holds true against a specified value, thus determining the significance of the coefficients and the overall constraints imposed on the model structure.

The function operates by comparing the performance of the full, unrestricted model against a restricted model where the hypothesized constraints are enforced. It requires the fitted model object, the definition of the hypothesis (or hypotheses), and the type of test (typically an F-test) as its key arguments, returning the resulting F-statistic, degrees of freedom, and the crucial p-value.

Introduction to Linear Hypothesis Testing in R

A linear hypothesis test is fundamentally important in econometrics and statistical modeling. It moves beyond simply examining individual p-values for predictors and allows for the testing of complex relationships or constraints among predictors simultaneously. For instance, a common application is testing the joint significance of a group of variables, effectively asking whether removing all these variables from the model would significantly degrade the model's explanatory power. This capability is crucial when dealing with models involving interaction terms, polynomial terms, or when theory dictates specific relationships between parameters.

When performing standard multiple regression, the output typically provides a t-test for each individual regression coefficient, testing whether that coefficient is individually different from zero. However, this method does not account for the joint influence or combined effect of multiple variables, especially in the presence of correlated predictors. The **linearHypothesis()** function addresses this gap by providing a mechanism to test multiple linear restrictions simultaneously, usually resulting in an F-statistic and an associated p-value for the overall set of restrictions. This joint testing procedure is often more robust than relying on a series of individual tests.

The function is designed for flexibility. While it is commonly used to test if coefficients are equal to zero (a test of exclusion), it can also be configured to test whether coefficients are equal to each other (e.g., $\beta_1 = \beta_2$) or whether they sum to a specific value (e.g., $\beta_1 + \beta_2 = 1$). The core principle remains the comparison of the full, unrestricted model against a restricted model where the specified hypotheses are imposed. The resulting test statistic quantifies the loss of fit incurred by imposing these linear restrictions.

Understanding the linearHypothesis() Function and Syntax

The **linearHypothesis()** function is integral for conducting robust statistical inference within R. It

requires three primary components: the fitted model object (the result of `lm()` or similar functions), the specific set of hypotheses to be tested, and optionally, the type of test (e.g., Type I, II, or III sums of squares, though Type III is often the default when testing joint restrictions).

This function is sourced from the powerful **car** package (Companion to Applied Regression), an essential library that extends base R's capabilities. Once the package is loaded, the function is ready to evaluate the specified linear constraints. The hypotheses themselves are typically passed as a vector of character strings, detailing the equations that define the null hypothesis.

The basic syntax for testing simultaneous constraints, using a fitted linear model named `fit`, is structured as follows:

```
linearHypothesis(fit, c("var1=0", "var2=0"))
```

This particular example illustrates a common use case: testing if the regression coefficients associated with variables **var1** and **var2** in the model are jointly equal to zero. Rejecting this joint null hypothesis implies that at least one of these predictors contributes significantly to explaining the variance in the outcome variable, even if their individual t-tests might suggest otherwise.

Prerequisites: The 'car' Package

Before executing the linear hypothesis test, it is mandatory to ensure that the **car** package is installed and loaded into the R session. This package provides specialized tools for regression diagnostics, including functions necessary for properly calculating restricted model sums of squares, which form the computational foundation of the **linearHypothesis()** function.

Installation is straightforward using the standard R command `install.packages("car")`. Following installation, the package must be activated using the `library(car)` command before the function can be called. Neglecting this crucial step will result in an error indicating that the function could not be found, preventing the execution of the test.

The reliability of the test statistic is highly dependent on the computational framework provided by the **car** package, which ensures the stability and correctness of the underlying F-test calculation, especially in complex scenarios involving interaction terms or unbalanced data structures.

Detailed Example: Setting Up the Regression Model

To demonstrate the utility of **linearHypothesis()**, we will walk through a practical example using a small dataset on student academic performance. The dataset includes information on the final exam score, the number of hours spent studying, and the number of practice exams taken for ten students. This setup allows us to explore how these two predictors jointly influence the outcome

score.

The first step in R is to define this dataset as a data frame, ensuring the variables are correctly structured for statistical analysis:

#create data frame

```
df <- data.frame(score=c(77, 79, 84, 85, 88, 99, 95, 90, 92, 94),  
hours=c(1, 1, 2, 3, 2, 4, 4, 2, 3, 3),  
prac_exams=c(2, 4, 4, 2, 4, 5, 4, 3, 2, 1))
```

#view data frame

df

score hours prac_exams

```
1 77 1 2  
2 79 1 4  
3 84 2 4  
4 85 3 2  
5 88 2 4  
6 99 4 5  
7 95 4 4  
8 90 2 3  
9 92 3 2  
10 94 3 1
```

We then proceed to fit a multiple linear model using the `lm()` function. Our model hypothesizes that the exam score is a linear combination of an intercept (`$beta_0$`), the hours studied (`$beta_1$`), and the practice exams taken (`$beta_2$`). The resulting object, named `fit`, contains all the necessary parameter estimates and diagnostic information.

#fit multiple linear regression model

```
fit <- lm(score ~ hours + prac_exams, data=df)
```

#view summary of model

```
summary(fit)
```

Call:

```
lm(formula = score ~ hours + prac_exams, data = df)
```

Residuals:

Min 1Q Median 3Q Max

```
-5.8366 -2.0875 0.1381 2.0652 4.6381
```

Coefficients:

```
Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept) 72.7393 3.9455 18.436 3.42e-07 ***
```

```
hours 5.8093 1.1161 5.205 0.00125 **
```

```
prac_exams 0.3346 0.9369 0.357 0.73150
```

```
---
```

```
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 3.59 on 7 degrees of freedom

Multiple R-squared: 0.8004, Adjusted R-squared: 0.7434

F-statistic: 14.03 on 2 and 7 DF, p-value: 0.003553

The standard summary output reveals that while **hours** is a statistically significant predictor (low p-value), **prac_exams** is not individually significant (high p-value). However, we must now formally test if these two factors, when considered together, are jointly unnecessary for predicting the score. This is where the joint hypothesis test becomes essential.

Implementing the Joint Hypothesis Test

Our goal is to formally test the most stringent null hypothesis regarding these two predictors: that the coefficient for **hours** and the coefficient for **prac_exams** are both simultaneously equal to zero. If we fail to reject this joint hypothesis, it implies that both variables could be omitted from the model without a statistically significant loss of explanatory power.

We execute the **linearHypothesis()** function, passing the fitted model `fit` and the vector of constraints, ensuring the **car** package is active:

```
library(car)
```

```
#perform hypothesis test for hours=0 and prac_exams=0
```

```
linearHypothesis(fit, c("hours=0", "prac_exams=0"))
```

Linear hypothesis test

Hypothesis:

```
hours = 0
```

```
prac_exams = 0
```

Model 1: restricted model

Model 2: score ~ hours + prac_exams

Res.Df RSS Df Sum of Sq F Pr(>F)

1 9 452.10

2 7 90.24 2 361.86 14.035 0.003553 **

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

The function generates a tabular output comparing the restricted model (Model 1, where the constraints are imposed) against the full model (Model 2). The test statistic is calculated based on the increase in the Residual Sum of Squares (RSS) when the constraints are applied, normalized by the change in degrees of freedom.

Interpreting the Output of linearHypothesis()

The output explicitly details the two models compared and the results of the F-test. Model 1 shows the RSS when the coefficients for hours and prac_exams are set to zero (effectively a model using only the intercept), while Model 2 shows the RSS of our full, estimated model. The difference in these RSS values, labeled "Sum of Sq," represents the variance explained by the two variables jointly.

The test results are summarized by the following critical values:

F test statistic: 14.035

p-value (Pr(>F)): 0.003553

These values relate directly to the defined hypotheses:

H0 (Null Hypothesis): Both regression coefficients are simultaneously equal to zero ($\beta_{\text{hours}} = 0$ AND $\beta_{\text{prac_exams}} = 0$).

HA (Alternative Hypothesis): At least one regression coefficient is not equal to zero.

Since the obtained p-value of 0.003553 is much smaller than the conventional significance level of $\alpha = 0.05$, we reject the null hypothesis. The statistical conclusion is that we have sufficient evidence to conclude that **hours** and **prac_exams**, taken together, contribute significantly to explaining the variation in exam scores. This finding supports the inclusion of at least one, and possibly both, of these variables in the predictive model.

Advanced Applications and Constraint Testing

The utility of **`linearHypothesis()`** extends well beyond merely testing if coefficients are zero. It is invaluable for testing complex, theory-driven restrictions. Consider the hypothesis that studying for an hour has the exact same impact on the score as taking one practice exam, meaning the two coefficients are equal ($\beta_{\text{hours}} = \beta_{\text{prac_exams}}$).

This constraint is tested by formulating the null hypothesis as a linear equality: $\beta_{\text{hours}} - \beta_{\text{prac_exams}} = 0$. The **`linearHypothesis()`** function handles this constraint by simply passing the algebraic expression as a character string:

```
linearHypothesis(fit, "hours - prac_exams = 0")
```

This flexibility allows researchers to test proportionality (e.g., $\beta_1 = 2\beta_2$) or additive effects (e.g., $\beta_1 + \beta_2 = 10$). By precisely defining the restrictions in this manner, **`linearHypothesis()`** ensures that the analysis is directly addressing the specific theoretical questions relevant to the research domain. Furthermore, the function can accommodate different types of test statistics (F-test vs. Chi-square), providing adaptability for various model types and sample sizes.